

DMlauncher documentation

COLLABORATORS

	TITLE : DMlauncher documentation		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		February 24, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DMlauncher documentation	1
1.1	DMlauncher documentation	1
1.2	Table Of Contents	1
1.3	Introduction	2
1.4	Installation	2
1.5	Usage	3
1.6	FILES	4
1.7	XPB	4
1.8	SIZE	4
1.9	NODATACHECK	5
1.10	PREFSFILE	5
1.11	The Prefsfile	5
1.12	History	6

Chapter 1

DMlauncher documentation

1.1 DMlauncher documentation

DMlauncher documentation

DMlauncher v1.4

© Alexis "Cyb" Nasr 1996-1997

\$VER: May 1997

[Table Of Contents](#)

[Introduction](#)

[Installation](#)

[Usage](#)

[The Prefsfile](#)

[History](#)

1.2 Table Of Contents

Table Of Contents

MAIN[DMlauncher documentation](#)

1.[Introduction](#)

2.[Installation](#)

3.[Usage](#)

3.1.[FILES](#)

3.2.[XPK](#)

3.3.[SIZE](#)

3.4.[NODATACHECK](#)

3.5.[PREFSFILE](#)

4.[The Prefsfile](#)

5.[History](#)

1.3 Introduction

1. Introduction

DMlauncher is part of the datamaster.library package & therefore should not be separated from it.

With this little program, you can associate a CLI commandline to each of the filetypes the library supports.

It is a CLI program, but it'll be much more powerful if you use it in association with ToolManager (making an Executable+Image+Icon object), it should boost your 'basic' workbench.... Just think of this: you'll just have drag & drop an icon on DMlauncher's & the most suitable action will be performed. Using it with FileMaster will give you a BIG boost too!

The program itself is 3 kb or so, and the library size depends of the number of external filetypes so I can't guess now, while typing this ;-)

Anybody wants to do a MUI based full-featured GUI? You're welcome! :)

NOTE:

(You'll find the assembler sourcecode of this program in the developer directory)

1.4 Installation

2. Installation

Needs

~~~~~

\* OS 2.04+

\* reqtools.library

Installation:

~~~~~

This is a CLI-tool that will surely go in C:

Copy the DMlauncher.prefs file in S: (and of course, modify it to suit your needs)

Toolmanager Installation

Well, if you really use Toolmanager to its best to boost your system, you surely know all that will follow, but anyway...

* Create an Executable-object with the CLI file

(set something like CON://///AUTO/WAIT/CLOSE for the output)

* Create an Image-object with the provided DMlauncher.Icon.

* Create an Icon-object associating both previous objects...that's it...

Now you have an icon sitting on your WB, you can just click it to have a Reqtools requester, or drag & drop icons on it. Nice no?

(I had a 'Multiview' object of this kind before... now I've replaced it with DMlauncher... Yeah!!... now I can drop anything on it, even musics ;-)

Toolmanager is (c) Stefan Becker (just great!)

reqtools.library is (c) Nico Francois

Filemaster 2 installation

"What? Filemaster 2? This old stuff? you must be kidding!" :-)

Yeah I know, now it's the age of memory hungry 1 Meg dir-utilities, like Directory Opus etc etc....

BUT I know there are lots of guys out there who stick to their good old 70Kb filemaster! *YES* !!

If you're a registered FM user, you have access to the config file. Then you can give a TERRIBLE BOOST to filemaster with DMlauncher!!!

Click on a DMS archive, it's unpacked to DF0:.... Click on a JPEG image, Visage displays it etc etc...

HOW? Well just edit your DMLauncher.prefs file, and create the following Command button in filemaster:

Command c:dmlauncher "{\$}" XPK

Name what you want

Pattern * <---- (or #?) so it will treat ANY file.

and that's all !!

Have fun with your "brand new" filemaster...

1.5 Usage

3. Usage

Typing the classic "DMlauncher ?" will give you the templates:

F=FILES/K/M,XPK/K/S,S=SIZE/K/N,NC=NODATACHECK/K/S,PF=PREFSFILE/K:

Explanation of the standard flags:

K means Keyword

S means Switch (works as a ON/OFF switch, no parameter needed)

M means multiple parameters possible.

FILES

XPK

SIZE

NODATACHECK

PREFSFILE

1.6 FILES

3.1. FILES

Well, just the file(s) that should be 'launched'...

You may use multiple files but keep in mind that the result will

STRONGLY depends whether the commands you execute are synchronous or not.

For example, if you select 4 IFF images, and the action is "run gfxviewer

%s"...well...you'll have 4 viewers being executed nearly in the same

time!!! Without the "run", of course the images 'd be loaded one after

another. To prevent this, you may also use the ALLFILES flag in the FLAGS

section of the definition, in [The Pref file](#).

If there are no files defined, a reqtools filerequester will pop up. You

may select multiple files, use patterns...

Note: VARIABLES

Each time a file is launched, the following GLOBAL VARIABLES will be defined:

DML_FILE : complete file path

DML_FILENAME : stripped filename

so you'll be able to get the filename if you defined an "execute script" 'complex' action.

When the requester is used, another variable 'll be defined:

DML_REQPATH

this is very useful as this path will be used the next time DMLauncher pops out a requester.

1.7 XPK

3.2. XPK

Very powerful option, as it XPK-loads the tested file (or part of it).

So all the 'XPK-packed' files will automatically be recognized correctly.

If you set this, be sure that the commands you run are able to xpk-unpack

the files you give them (make them xdata or MCP clients if not, easy)

(off by default)

1.8 SIZE

3.3. SIZE

By default, the program loads 5000 bytes of the file for test. This is enough for all tests. You may change this value (in bytes) if you want. Setting it to -1 will load the WHOLE file.

1.9 NODATACHECK

3.4. NODATACHECK

When the file is tested DMlauncher will first check the file by DATA (most powerful mode, but a bit slow as it needs to load some/all data of the file). Then if the type is still 'generic', it will use PATTERN checking as last chance.

adding NODATACHECK will force the test to use ONLY the pattern-matching check on the filename, which is quicker... but not always very precise.

(best example is ASCII files without extensions ;-)

1.10 PREFSFILE

3.5. PREFSFILE

Default is S:DMlauncher.prefs

You can have multiple Prefsfiles, that's handy if you want to adapt the actions to particular environments.

You can also use this to store the prefsfile in ENVARC: instead.

The advantage is that it will be copied to ENV: on startup.

Then just set PREFSFILE="ENV:myfile" when executing dmlauncher.

Advantage? being in RAM:, the prefsfile will be loaded very fast...

1.11 The Prefsfile

4. The Prefsfile

Each filetype should be associated to a particular action. Don't worry, if it happens that you drop a file that has no associated action, DMlauncher will warn you & ask you if you want to use the 'generic' type action.

(everything is case insensitive)

Each item consists of:

TYPE=filetype

This can be a MAJOR-Filetype like "MUSIC-Synth" or a SUB-Filetype like "SoundMon 2.2". As you like!

ACTION=just enter a command line exactly like from the CLI.

The %s string will be replaced by the filename.

As CLI tools react badly to filenames containing spaces, you must type

"%s", not just %s.

There is ONE exception: if the ALLFILES flag is set, use %s instead as
 DMlauncher will then automatically put EACH filename between ''')
 FLAGS=....

That's an optional line. The only available flag for the moment is ALLFILES :
 Use it for commands that accept multiparameters. I've added this, thinking of
 tools like Visage or Fastview, that really gain much, being given all the
 files, as they will load next image while you're watching the current one.
 So, if a checked file is associated to a tool with this ALLFILES
 flag, then ALL the other files will be passed to this tool. The drawback of
 this, is that all files should be of the same type. It's logical but it can
 be quite tempting to pass a mix of JPEGs & GIF to DMlauncher, the problem
 is that if you defined...

TYPE=GFX-GIF

ACTION=SYS:Utilities/Fastview %s

FLAGS=ALLFILES

END_DEF

TYPE=GFX-JPEG

ACTION=SYS:Utilities/FJPEG %s ;specific

END_DEF

...& the first file is a JPEG, then ALL files will be passed to
 FJPEG (the GIFs won't work, that's the annoying thing ;-)

Anyway, all this never happens with DirOpus or Dirwork tools, as they only
 provide 'double-click direct actions' for ONE file... If DMlauncher has
 these little problems, it's only because... it allows you to choose
 multiple files in the same time!! Not so bad, is it? hehehe.
 END_DEF : this ends the definition. Do NOT forget it eh?

Look at the example prefsfile, it'll make things clearer :)

1.12 History

5. History

v1.4 (v2.0 library)

~~~~~

- \* supports sub-filetypes
- \* corrected a stupid heavy bug (when closing reqtools requester->crash).
- \* minor bugs cleaned.
- \* better handling of ''''s in command line (there are really a mess! :)

v1.13

~~~~~

* BUGFIX:

In the initial versions of DMlauncher, the program terminated after having 'launched' some action via the Reqtools file requester. Then I thought it'd be much nicer if the requester popped up again & again. So I did, & it worked fine... Except that I had forgotten to clear some flags !! Before, that was no problem as they were used ONCE! But now, these flags caused some mess in some circumstances (typically double memory-freeing=>GURUS!). This is fixed now. (sorry !!)

v1.0 -> v1.12 (pre-release)

~~~~~

This is a very tiny program, that was mostly made as an example... Anyway tools that do the same thing already exist, like DropIt or the excellent

**\*\*ClassAction\*\***. (Even if DMlauncher is FAR smaller in size :)

I could add some lacking features like the multi-action selection (show, edit, print etc...), selectable WB/CLI/REXX launching ...

For the moment I don't feel the necessity of it, as I use DMlauncher as a quick'n'small show-only tool, when it's not worth running a big Dir-tool.

It may be done if I receive a sufficient amount of user-requests :-)